Evaluation of Internal Models in Autonomous Learning

Evaluation of Internal Models in Autonomous Learning

Simón C. Smith and J. Michael Herrmann

***Abstract*—Internal models (IMs) can represent relations between sensors and actuators in natural and artificial agents. In autonomous robots, the adaptation of IMs and the adaptation of the behaviour are interdependent processes which have been studied under paradigms for self-organisation of behaviour such as homeokinesis. We compare the effect of various types of IMs on the generation of behaviour in order to evaluate model quality across different behaviours. The considered IMs differ in the degree of flexibility and expressivity related to, respectively, learning speed and structural complexity of the model. We show that the different IMs generate different error characteristics which in turn lead to variations of the self-generated behaviour of the robot. Due to the trade-off between error minimisation and complexity of the explored environment, we compare the models in the sense of Pareto optimality. Among the linear and nonlinear models that we analyse, echo-state networks achieve a particularly high performance which we explain as a result of the combination of fast learning and complex internal dynamics. More generally, we provide evidence that Pareto optimisation is preferable in autonomous learning as it allows that a special solution can be negotiated in any particular environment.**

***Index Terms*—Autonomous robot, internal model, prediction error, homeokinesis, time-loop error.**

I. Introduction

An internal model (IM) represents various relationships between the state of a robot, its actions and the stream of sensory input it receives. More specifically, forward models use the initiated motor command to produce an estimate of subsequent sensory input, whereas inverse models can determine an action that leads to a desired state. IMs are essential in many control architectures for robots and are relevant also in biological motor control [1]. Experiments have provided evidence for the existence of internal models in animals and humans [2], [3], e.g. the precise control of an object gripped by index finger and thumb is enabled by a system that comprises both a forward and an inverse model of the arm [4]. In control theory, the Smith predictor, for example, combines models to improve feedback control by compensating for slow responses and weak power gain. Also, IMs have been regarded as indispensable for delayed feedback control [5], and are considered as useful for state estimation, confirmation and cancellation of sensory input, and context estimation [6].

In the present paper, we aim at evaluating the quality of IMs in autonomous learning, i.e. in the case where no specific goal is given or none of the possible goals can be reached by the available policies. In this situation, an agent needs to find a new policy and to perform some form of exploration. The connection between exploration and curiosity has been discussed already in the 1960s [7] and was later recognised as an important aspect of autonomous learning [8]–[10] and constitutes a practically applicable form for the idea of dual control [11]. Theoretical progress was achieved by the more recent information-theoretic studies [12]–[15] that enable a direct comparison of the information gain by exploration and the loss of predictability of the current behaviour, if the relevant entropies can be sufficiently well estimated.

In the present context, IMs are used in order to predict sensory input based on motor output, while the generation of exploratory behaviour is based on one of the standard methods (s. Sect. II-A). When facing new sensory information, the agent will typically experience a difference between its prediction and the new evidence. This discrepancy can be used either to reduce the perceived error by aiming at obtaining similar data or to decide to move to other regions of the environment where smaller errors are more easily achievable. These errors can be due to unreliable sensors, inaccurate state estimation, imprecise actuators or stochasticity of the environment. As the agent may not be able to identify the causes of the error nor alleviate them during a mission, we assume that its goal is merely to obtain as much information as possible in a given situation. This simplified setting allows us to focus on the question whether the agent should prefer to represent within its internal model a small part of the environment exactly or a large part with less accuracy.

Consider the idealisation of a robot that is able to predict all future states perfectly. There is thus neither a necessity not a possibility to learn (if we restrict ourselves to error-driven learning). In this idealisation — which is in fact the assumption underlying open-loop control — the robot has full information and can proceed to execute its task without the need of any exploration. Practically, however, the discrepancies between internal estimates and external data will provide the information that is critical for learning and behaviour. We will thus ask whether a nearly perfect robot is learning very little, while a robot that perceives larger errors can learn more efficiently.

Our results suggest that the choice between the low- and high-error regime is largely independent of the type of the IM, but is made within the interaction of the agent with its environment. In order to be able to compare the performance of agents with different IMs, we were lead to consider the learning task as a Pareto optimisation problem. This seems appropriate also because many approaches to autonomous learning [8], [13]–[18] are based on the joint optimisation of

Simón. C. Smith and J. Michael Herrmann are with the Institute of Perception, Action and Behaviour, School of Informatics, The University of Edinburgh, 10 Crichton St, Edinburgh, EH8 9AB, U.K.
E-mail: artificialsimon@ed.ac.uk, michael.herrmann@ed.ac.uk

Manuscript received XX/XX, XXXX; revised XX/XX, XXXX.

two incompatible objectives such as sensitivity and predictability [16], see Sect. II-A. Pareto optimisation [19] refers to the solution of a problem with several goals where the pursuit of one goal should not lead to the deterioration with respect to any of the other goals. In other words, any combinations of the goals is optimal in the Pareto sense if it is not *dominated* by another solution, i.e. if there is no solution that is at least equal with respect to all goals and better with respect to at least one of the goals. The set of the known non-dominated solutions of a multi-objective optimisation problem is called the Pareto frontier. For autonomous agents, the balance between the goals (e.g., exploration and prediction quality) cannot be specified beforehand and may also vary during run-time such that different agents should not be evaluated based on a fixed combination of the goals. Instead, we will compare the Pareto frontiers found by the agents.

Indeed, we observe a tendency to choose different positions along the Pareto frontier, e.g., in dependence on the complexity of the environment, but the Pareto formulation provides us with a method to rate the quality of the internal model without reference to the stage of the exploratory phase and the environmental complexity. In other words, we will study the quantitative relation between the types of errors experienced by the robot and their effect on the generation of behaviour.

The rest of the paper is organised as follows: In Section II-A we will explain the control scheme that brings about exploratory behaviour based on the above-mentioned discrepancy principle mentioned above. Section II-C describes the types of internal models to be considered in this context. The experimental results in Section III will provide evidence for the Pareto formulation of the autonomous learning problem. Finally, Section IV will discuss the implications and potential generalisations of the presented approach.

II. Methods

A. Homeokinetic control

Homeokinesis [20] is an unsupervised active learning algorithm that shapes the interaction between a robot and its environment. By updating the parameters of the controller, i.e. of the sensor-to-action map, the robot aims at a balance between predictability of future inputs and sensitivity with respect to current inputs. The resulting behaviour arise from random fluctuations and develops temporally coherent and correlated motion across multiple degrees of freedom, although the sign of the correlation may change in time. Homeokinesis uses an IM as a predictor of future input signals. The predicted input is compared with the sensory input giving rise to the *prediction error*. This error is then used to modify the behaviour of the robots as well as to improve the predictive model. For the task of learning multiple goal-directed behaviours simultaneously [21], IMs are also used as predictive models in conjunction with a continuous-time recurrent network (CTRNN) in order to learn to regenerate sensory sequence patterns. Different modalities of sensation (vision-based object position and arm joints proprioception) are taken as input, then the CTRNN fuses these inputs to generate prediction of their time developments in the future. The network is trained to minimise the error between the teaching sequence pattern given from the outside and the predicted sequence pattern generated by itself. Then, the predicted sensory sequences are sent in a closed-loop configuration to the input, generating the behaviour of the robot given an initial context. Robots can develop an IM to rely on it to complete their task when sensory stimulation is temporarily unavailable [22]. It has been shown that a simulated robot evolve to display navigation skills when the actual sensory input is deprived, using its IM by anticipating functional properties of the next sensory state rather than the exact state that sensors would have assumed. According to [23], curious agents are interested in learnable but yet unknown regularities, and get bored by both predictable and inherently unpredictable interactions. Mismatch between reality and expectations are translated into curiosity rewards for curious, creative, exploring agents that like to create or observe surprising aspects of the world in order to learn novel patterns. The next input is predicted by a data compressor, an IM, using some history of actions and inputs. The action-generating, reward-maximising controller get rewarded for action sequences provoking still unpredictable inputs. To discourage the controller from focusing on truly unpredictable, random inputs, the expected progress of the predictor is modelled: parts of the world where the data compressor fails to learn. Another scenario is when a robot is able to indirectly infer its own morphology through self-directed exploration and then use the resulting self-models to synthesise new behaviours [24]. If the robot's topology unexpectedly changes, the same process restructures its internal self-models, leading to the generation of qualitatively different, compensatory behaviour.

The homeokinetic controller is a parametric function

$$\mathbf{y}_t = K(\mathbf{x}_t; C) \tag{1}$$

of the vector $\mathbf{x}_t$ of current sensory states of the robot and generates a vector of motor commands $\mathbf{y}_t$ in dependence on the current values of the parameter matrix C. The update of the parameters is based on a comparison of actual inputs and their prediction by means of an internal model. This model

$$\hat{\mathbf{x}}_{t+1} = M(\mathbf{x}_t, \mathbf{y}_t; A), \tag{2}$$

produces a prediction of future states $\hat{\mathbf{x}}_{t+1}$ based on the current input $\mathbf{x}_t$, action $\mathbf{y}_t$, or both. The difference between actual and estimated state defines the prediction error

$$\boldsymbol{\xi}_{t+1} = \mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}, \tag{3}$$

which gives rise to one of the two objective functions that are used here:

$$\mathcal{E}_{t+1} = \boldsymbol{\xi}_{t+1}^{\top} \boldsymbol{\xi}_{t+1}. \tag{4}$$

The squared prediction error $\mathcal{E}$ enters the update rule for the the parameters A of the model (2) as a sliding average with a short time scale.

Inserting Equation 1 into Equation 2, we can consider the model as a function defined on the state space:

$$\psi(\mathbf{x}_t) = M(\mathbf{x}_t, K(\mathbf{x}_t); A), \tag{5}$$

which together with (3) defines a dynamical system that represents the trajectory of the robot $\mathbf{x}_{t+1} = \psi(\mathbf{x}_t) + \boldsymbol{\xi}_{t+1}$.

It is of importance in homeokinetic learning to define an input shift $\boldsymbol{\eta}$ corresponding to the error $\boldsymbol{\xi}$. It is given by

$$\boldsymbol{\eta}_t = \arg\min_{\boldsymbol{\eta}} \|\mathbf{x}_{t+1} - \psi(\mathbf{x}_t + \boldsymbol{\eta})\|^2 \tag{6}$$

or, if ψ is invertible, simply as $\boldsymbol{\eta}_t = \psi^{-1}(\psi(\mathbf{x}_t) + \boldsymbol{\xi}) - \mathbf{x}_t$. Using a Taylor expansion, $\psi(\mathbf{x}_t + \boldsymbol{\eta}_t) = \psi(\mathbf{x}_t) + L(\mathbf{x}_t)\boldsymbol{\eta}_t + O(\|\boldsymbol{\eta}_t\|^2)$, we can express the prediction error (3) in linear order as $\boldsymbol{\xi}_{t+1} = L(\mathbf{x}_t)\boldsymbol{\eta}_t$, where

$$L = (\partial\psi_i(\mathbf{x})/\partial x_j) \tag{7}$$

is the Jacobian matrix of the system. Using Equation 6 or, if the inverse of L exists, $\boldsymbol{\eta}_t = L_t^{-1}\boldsymbol{\xi}_{t+1}$, we define the *time loop error* as:

$$\mathbf{E}_t = \|\boldsymbol{\eta}_t\|^2 = \boldsymbol{\eta}_t^\top \boldsymbol{\eta}_t = \boldsymbol{\xi}_{t+1}^\top (L_t L_t^\top)^{-1} \boldsymbol{\xi}_{t+1}, \tag{8}$$

which is the second error functions in the homeokinetic controller. The homeokinetic learning rule updates the parameters C of the controller (1) by gradient descent

$$\Delta C = -\varepsilon_C \frac{\partial \mathbf{E}_t}{\partial C}, \tag{9}$$

where ε_C is a learning rate. Because $\mathbf{E}_t$ depends[1] on the model M, the learning rule (9) will have different forms for different models. Below, we will derive the learning rules for a linear model. For the other models, to be introduced in Section II-C, see Appendix A.

B. Controller update for a linear predictor

In order to achieve fast adaptation of the controller (1), it is tempting to use a quasi-linear controller

$$\boldsymbol{y}_t = K(\mathbf{x}_t) = g(C\mathbf{x}_t + \mathbf{h}), \tag{10}$$

where g is an element-wise sigmoidal function and the adaptive bias term $\mathbf{h}$ can be considered as a row of the matrix C corresponding to a constant input line. The choice of a linear model

$$\hat{\mathbf{x}}_{t+1} = M(\mathbf{y}_t) = A\mathbf{y}_t + \mathbf{b}, \tag{11}$$

may limit the complexity of representable sensorimotor maps, but is interesting in comparison to other choices considered below. The parameters of the controller and the model are now the matrices C and A, respectively, which are complemented by the corresponding bias vectors $\mathbf{h}$ and $\mathbf{b}$.

Because of the simple structure of (10), we can omit here the state dependence (2) and define the model M only in motor space. The model defines the dynamics (5) $\psi(\mathbf{x}) = Ag(C\mathbf{x} + \mathbf{h}) + \mathbf{b}$ and the Jacobian (7) can be obtained explicitly as

$$L(\mathbf{x}) = AG'C, \tag{12}$$

where G' is a diagonal matrix with nonzero entries $g'(C\mathbf{x}+\mathbf{h})$. The parameter update (9) becomes

$$\Delta C = \varepsilon_C \; \boldsymbol{\eta}^\top L \frac{\partial L}{\partial p} \boldsymbol{\eta}, \tag{13}$$

[1]The dependency is both via $\boldsymbol{\xi}$ and L that is usually also derived from the model rather than from the actual dynamics.

and analogously for the bias term $\mathbf{h}$. With $\boldsymbol{\mu} = G'A^\top \left(L^\top\right)^{-1} \boldsymbol{\eta}$ and $\boldsymbol{\zeta} = C\boldsymbol{\eta}$ the learning rules for a linear controller with a linear model are

$$\Delta C_{ij} = \varepsilon_C \mu_i \eta_j - 2\varepsilon_C \mu_i \zeta_i y_i x_j \tag{14}$$

$$\Delta h_i = -2\varepsilon_C \mu_i \zeta_i y_i, \tag{15}$$

with i and j representing indices of the matrices and vectors. Simultaneously, but possibly with a different learning rate ε_A, the parameters of the linear model (11) are updated via gradient descent on the standard prediction error (4).

$$\Delta A_{ij} = -\varepsilon_A \frac{\partial \mathcal{E}}{\partial A_{ij}} = \varepsilon_A \xi_i y_j \tag{16}$$

$$\Delta b_i = -\varepsilon_A \frac{\partial \mathcal{E}}{\partial b_j} \; = \varepsilon_A \xi_i \tag{17}$$

C. Algorithms for internal models

In order to assess the controller behaviour in dependence on the model, two classes of algorithms for model learning were tested, namely linear regression analysis and (nonlinear) recurrent neural networks. In the present context the model does not have to converge, i.e. although a global relation between inputs and outputs may be desirable, it also possible that a flexible model follows the non-stationary statistics generated by the robot's behaviour. The latter would lead to a series of local models which may be stitched together in an off-line stage, while the former typically requires prior knowledge about the robot and the environment. Our goal is not to find out which approach is preferable in what situation, but to show that both goals are connected as a Pareto optimisation problem.

1) Linear models: Regression methods aim at the estimation of the parameters of a hyperplane in the training data space in order to describe future data. Linear regression may not provide the expressivity needed in a complex environment, but it has the flexibility to follow the changes in the behaviour of the robot. We consider the effect of gradient-based adaptation of the model as compared to a recursive least-square fit, and include a locally weighted regression as a more expressive variant.

These three methods share the characteristic that the components of the state and input signal $\mathbf{z}_t^\top = [\mathbf{x}_t^\top \; \mathbf{y}_t^\top]$, are linearly combined to produce the desired output.

$$\hat{x}_{t+1} = \sum_{i=0}^{m} a_i z_i + b. \tag{18}$$

In matrix notation, this can be written as $\hat{\mathbf{x}}_{t+1} = A\mathbf{z}_t + \mathbf{b}$. Compared to Equation 11, we now include state dependency.

The first variant, iterative linear model (ILM), updates the parameter by gradient descent over the prediction error, see (16, 17). For the second variant (LReg), recursive least squares are used to fit the parameters at each new input output pair arrival:

$$A_{t+1} = A_t + P_{t+1}\mathbf{z}\boldsymbol{\xi}^\top,$$

$$P_{t+1} = P_t - \frac{P_t \mathbf{z}\mathbf{z}^\top P_t}{1 + \mathbf{z}^\top P_t \mathbf{x}}.$$

The derivation of the recursive rule is based on the Woodbury matrix identity [25] and the matrix P is initialised as a diagonal matrix with large entries.

The third method, locally weighted regression (LWR) [26], includes a diagonal weight matrix W that accounts for the distance between the input and the data stored so far. The motor commands and sensory data vectors are collected for a number ω of time steps as rows of the matrices Z and X.

$$W_{ij} = \begin{cases} e^{-\frac{\|Z_{ij} - \mathbf{z}_i\|}{2\sigma}} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The parameter σ is adjusted based on the standard deviation of the data. The regression parameters are obtained by

$$A = (Z^\top W Z)^{-1} Z^\top W X. \tag{19}$$

For on-going learning, not all sensor and motor data can be stored. Thus, only the last $\omega \approx 2,000$ values (corresponding to the last 20 seconds in simulated real time) are used.

2) Nonlinear models: We also consider nonlinear models, namely artificial neural networks, where we limit ourselves to two variants: recurrent neural networks (RNNs) trained by real time recurrent learning and echo-state networks[2]. The current state of an output neuron of an RNN depends also on the sequence of previous input. These properties appear to be appropriate as a model of the interaction of the robot with the environment such that the robot can in principle react differently upon the same current input.

In addition to RTRL and ESNs, we have tested other learning algorithms for RNNs such as backpropagation through time (BPTT) [27]. Results have been omitted here because they were not competitive with the above approaches.

a) Real time recurrent learning (RTRL).: Real time recurrent learning [28] is an efficient training algorithm for RNNs. The weights of the connections between neurons are updated continuously by the arriving data such that the methods is suitable for on-line learning in autonomous robots. In the present context, an RTRL network receives the motor commands (1) and actual sensor states as inputs, and generates a prediction of the next state (2). The connection weights are updated by gradient descent on the squared prediction error (4):

$$\Delta W_{ij} = -\varepsilon_A \frac{\partial \mathcal{E}}{\partial W_{ij}} = \varepsilon_A \sum_{k \in O} \xi_k(t) p_{ij}^k(t),$$

$$p_{ij}^k(t+1) = g'(s_k(t)) \left[\sum_{l \in H} W_{kl} p_{ij}^l(t) + \delta_{ik} v_j(t) \right],$$

where $\xi_k(t)$ is the prediction error (3) at the k-th output neuron, $v_j(t)$ is the activation of neuron j at time t, $s_j(t)$ is the sum of weighted inputs to this neuron, H is the set of hidden neurons and O is the set of output units for which an error is directly defined. The term $p_{ij}^k(t)$ is initialised by $p_{ij}^k(t_0) = 0$ and represents the sensitivity of the output nodes to a change of the weights. Note that the weight W_{ij} is not necessarily connected with the output unit that predicts $\hat{x}_{t+1,i}$, thus this learning rule is non-local.

Even though RTRL requires less computational resources than back-propagation through time, $O(n^4)$ vs. $O(n^3)$ complexity, it is still computationally expensive and requires large storage capacity. The derivation of the response model for RTRL can be found in the Appendix B.

b) Echo-state network (ESN).: In an echo-state network [29], improved state predictions are obtained by adjusting linear weights from a recurrent neural reservoir network and from the input layer to the output units. The connections within the reservoir and feedback from the output layer are not subject to adaptation, except that its major eigenvalue is adjusted to a near-critical value as part of the initialisation of the network.

The prediction of next input is obtained by:

$$M(\mathbf{x}_t, \mathbf{y}_t) = \hat{\mathbf{x}}_{t+1} = W^{\mathrm{xo}} \mathbf{x}_t + W^{\mathrm{yo}} \mathbf{y}_t + W^{\mathrm{so}} \mathbf{s}_t, \tag{20}$$

where $\hat{\mathbf{x}}_{t+1}$, $\mathbf{x}_t$ and $\mathbf{y}_t$ are the same as defined above. The vector $\mathbf{s}_t$ represents the reservoir state at time t, the matrices W^{so}, W^{xo} and W^{yo} represent the connections weights, respectively, from the reservoir to the output, from input to output and from the motor commands to the output. The update of the reservoir state is defined as

$$\mathbf{s}_{t+1} = g_{\mathrm{ESN}}(W^{\mathrm{xs}} \mathbf{x}_{t+1} + W^{\mathrm{ys}} \mathbf{y}_{t+1} + W^{\mathrm{ss}} \mathbf{s}_t), \tag{21}$$

where the matrix W^{ss} represents the weights within the reservoir, W^{xs} are the weights from the sensors state to the reservoir and the matrix W^{ys} represents the weights from the motor commands to the reservoir. The function g_{ESN} is a sigmoidal function applied element wise.

A gradient descent algorithm, in order to minimise the prediction error (3), is used to update the weights connected to the output $W = W^{\mathrm{so}} \oplus W^{\mathrm{xo}} \oplus W^{\mathrm{yo}}$. The update rules are

$$\Delta W_{ij} = -\varepsilon_A \frac{\partial \mathcal{E}}{\partial W_{ij}}. \tag{22}$$

3) Noise controller: For comparison, we also include a trivial controller. Coloured noise is used as the parameters of the controller (1). A time average

$$\tau c_{t+1,i} = -c_{t,i} + \sqrt{\tau} n_t, \tag{23}$$

with rate τ is used to determine the colour of the noise. The variable $c_{t,i}$ denotes the parameter i of the controller at time t, and n_t is a normally distributed random variable at time t. For $\tau \ll 1$, the coloured noise is close to white, while for bigger values of τ, strongly coloured noise is produced. The temporal correlation of the noise is fixed to match the average correlation in the other cases. A model is not needed in this case.

D. Experimental setup

Experiments were conducted using a simulated four-wheeled robot on an undulating surface, see Figure 1. The hills were placed randomly in the environment with widths similar to the size of the robot. The height of the hills was varied in the experiments by scaling the vertical coordinate

[2]Also, we tested other nonlinear models. We found similar behaviour to the RTRL model, see Section IV.

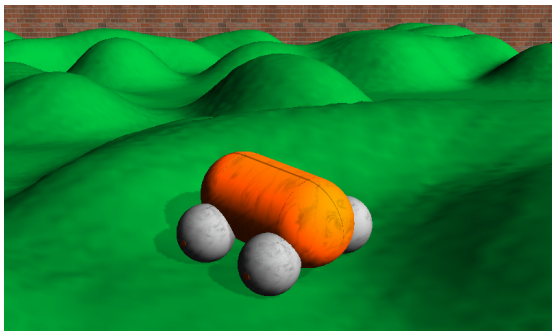
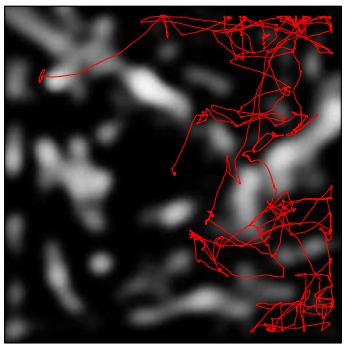


Figure 1: A four wheeled robot simulated in the LpzRobots robot simulator [30]. The robot has proprioceptive sensors for the angular velocity of the wheels in direct correspondence to the control output. Because of conflicts among the wheels, friction and slope, commanded and measured angular velocity can differ.

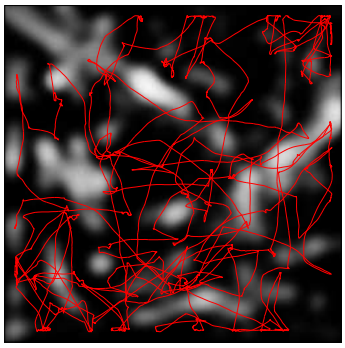
in order to generate several levels of difficulty. The scaling levels vary from zero (flat surface: difficulty 0) to a maximal slope (difficulty 1) which was defined by the limits of the motor forces of the robot, i.e. if the robot had been positioned ideally and used an ideal controller, it would be capable of climbing all slopes. Note that this ideal controller may or may not be found by the different versions of the learning control algorithm used here, and that the peaks may be accessible also by detours or using momentum. In this way, exploration is not limited by the environment, but only by the flexibility and capability of the models. The environment was enclosed by a squared-shaped wall with side lengths equal to 20 times the length of the robot.

The level of exploration of the robot was measured by the coverage of the environment within a single trial of a duration of 20 minutes of simulated real time. For this purpose, the accessible area was partitioned into 10×10 square patches. The partition has an evaluation purpose and does not affect the behaviour or internal models of the agent. While a higher definition of the partition will increase the number of bins, we have chosen the bin size to be the twice as of the robot. This size accounts for one complete movement from one bin to another to be counted. Full coverage meant that the robot traversed all boxes at least once (Fig. 2). For comparability across different difficulty levels, coverage is considered with respect to the planar environment, i.e. the height dimension is ignored for the evaluation. In addition, we also consider the length of the path measured as the number of cells the robot entered, but now including repetitions.

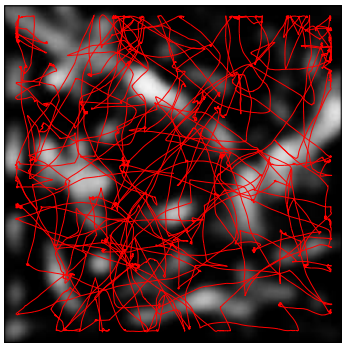
These quantities were recorded and compared to the prediction error that was simultaneously obtained from one of the internal models. Learning rates were chosen for each model in order to optimise the performance with respect to a Pareto optimum of prediction error and coverage, see Section III.



(a) 40% of coverage



(b) 85% of coverage



(c) 95% of coverage

Figure 2: Sample trajectories produced within 20 minutes of simulated real time. Coverage of the environment is measured as a fraction of visited cell in an virtually overlaid grid and depends on the properties of the internal model of the agent. Lighter colour in the images represents higher elevation in the environment.

III. Experimental results

A. Prediction and mobility

Prediction errors depend on the behaviour of the robot as much as on the properties of the environment. The increase of the difficulty of the environment leads to an overall decrease of the prediction error for most of the IM training algorithms tested, see Figure 3. This seems to contradict the expectation that a landscape with stronger undulation causes more disturbances to the robot behaviour, but will become clear when we consider the fraction of the environment that was actually visited by the autonomous robot. Each data point in the plot represent an average over a series of trial for a set of specific learning rates. For most of the models, the prediction error tends to decrease with larger difficulty of the terrain.

At higher difficulty, the robot tends to cover a smaller part of the arena (see Figure 4), which may be due to various

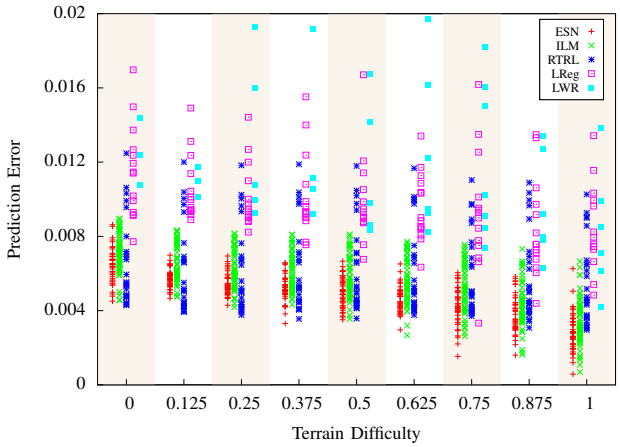


Figure 3: Prediction error for all models at different terrain difficulty. Each data point represents an average over a series of trials for a specific learning rate such that the lowest values in each column represent optimal rates. ESN and ILM tend to show best results, while the performance of the RTRL network appears to depend less strongly on the terrain difficulty.

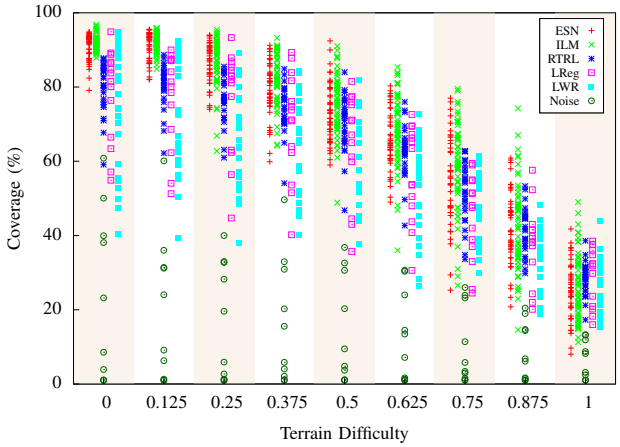


Figure 4: Coverage percentage for different terrain difficulty with several learning rates for the models and controller. At lower terrain difficulty virtually, full coverage is attained. At higher terrain difficulty, ESN and linear models have the best performance. The coloured-noise controller achieves only poor performance for any τ (23).

reasons: (i) The current controller parameters may often not be in a region where enough motor power is generated to climb steep slopes. (ii) For a fast learning model, the errors may be too small to switch behaviours as necessary to manoeuvre in the more difficult landscape, which can be due to the lack of complexity of the internal model. (iii) At higher difficulty, a larger error in a model with slower learning speed leads to more frequent behavioural changes such that the robot lacks the coherency necessary to climb, i.e. although the controller parameters change quickly, the robot's exploratory movements are limited to local exploration, i.e. the robot merely moves back and forth and thus does not cover much of the terrain. Reasons (ii) and (iii) can be interpreted as the impossibility of

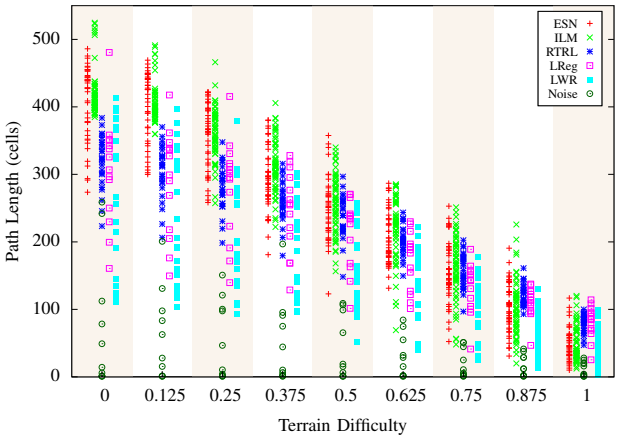


Figure 5: The total path length covered by the robot at different terrain difficulty. The ILM and ESN models have the best performance. The linear regression and the LWR models show comparable performance for very few combinations of different learning rates. Coloured noise has poor performance for the majority of τ values.

a full coverage of the arena by a homeokinetically controlled agent at higher terrain difficulty. In order to find and maintain a specific motor activation, the controller learning rule (Eqs. 14, 15) requires small prediction error and a sufficiently complex internal model. Such a configuration is possible, but because the controller is not explicitly favouring steep slopes, the easier parts of the environment are more likely to be explored. In fact, homeokinesis tends to withdraw from situations that diminish sensitivity, i.e. the agent tends to avoid steep slopes even if they are surmountable in principle. If, however, the prediction error becomes low, e.g. when the robot remains within a small region, then the homeokinetic controller will increase its exploratoriness according to the balance implied by Eq. 8.

In respect to the differences among the models, the coverage results (Figure 4) are similar to the prediction results (Figure 3), i.e. the ESN and the linear approaches perform best by achieving a higher coverage, but the RTRL network does not follow the general trend of a reduced coverage at hillier terrain. As a baseline, we also included a noise controller. The noise controller shows that the exploratory behaviour is enhanced by the homeokinetic controller, see Figure 4. This seems to indicate that exploration requires a correlation between the learning signals and the interaction of the robot with the environment. For a flat or nearly flat ground, almost full coverage is obtained by most of the models, except for the noise controller. Only with an optimised correlation in the noise generator, the noise controller results similar to some of the less good models.

Another way to measure the performance of the controller is the length of path travelled by the robot, see Figure 5. Path length is estimated by the number of times the robot crosses a boundary between two cells of the 10×10 grid that is overlaid to the arena. The ILM and the ESN lead to a larger path length in most cases. As expected, with a higher difficulty the total path length decreases.

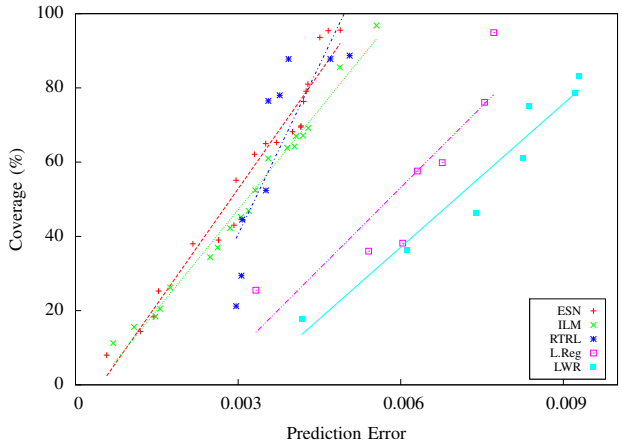


Figure 6: Pareto frontier of the optimal combination of maximum coverage vs. minimum prediction error. ESN responds with higher coverage and lower prediction error for several learning rates. The ILM has a comparable coverage percentage but with higher prediction error. The rest of the models achieve good coverage but at the expense of a larger prediction error.

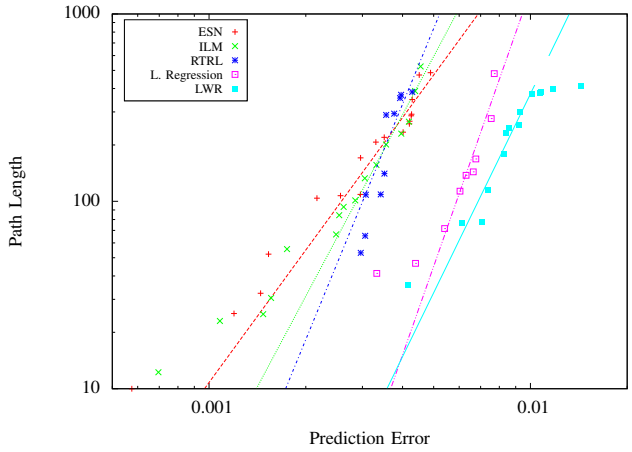


Figure 7: Path length of the exploration vs. the prediction error. To the left the two best performances achieved by the ESN and the ILM. The rest of the models produce similar path lengths but at higher prediction error. Curves are obtained by exponential regression.

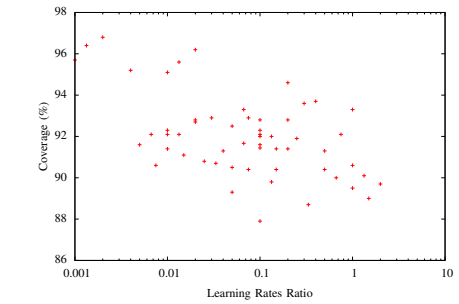
B. Optimal learning rates

The previous measures indicate a trade-off between mobility of the robot and prediction error. Indeed, the scatterplot (Figure 6) indicates a linear relationship, namely that higher mobility occurs when the prediction error is also high when we restrict ourselves to optimal learning rates. The optimality of the learning rate can now be defined as a Pareto optimum for the simultaneous minimisation of error and maximisation of coverage (or similarly path length, see below).

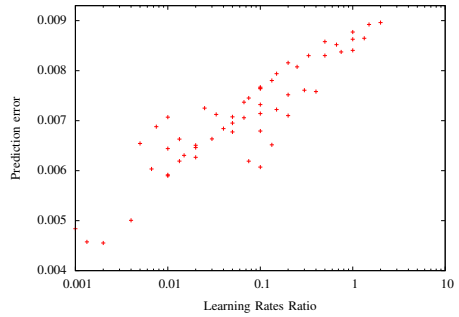
ESN finds generally the best trade-off between the two characteristics. At larger errors, RTRL leads to an advantage on coverage, while at small errors the ILM achieves a performance comparable to ESN.

	model type	model complexity	exploration	prediction	environment complexity	performance
linear	ILM	−	0	+	−	+
	LWR	−	−	−	0	−
	LReg	−	−	−	0	−
nonlinear	RTRL	+	+	0	+	+
	ESN	+	+	+	+/−	++
	noise	−	0	n/a	0	n/a

Table I: Overview of the results for the different models, see Sect. II-C. Entries "+" refer to high, "−" to low, and "0" to medium values. "+/−" indicates that all values are suitable.



(a) Coverage



(b) Prediction error

Figure 8: Coverage and prediction error compared to the ratio of the learning rates ($r = \varepsilon_C/\varepsilon_A$). Both best results, high coverage and low prediction error, are obtained at a small ratio, i.e. at model learning rate larger than controller learning rate.

Similarly, path length vs. prediction error are plotted in Figure 7. ESN and ILM models achieve smaller prediction errors while retaining comparable path lengths. On the other hand, larger path are achieved by the ILM, ESN and RTRL with similar prediction error. The other models have comparable path length but at a higher prediction error. The results are summarised in Table I.

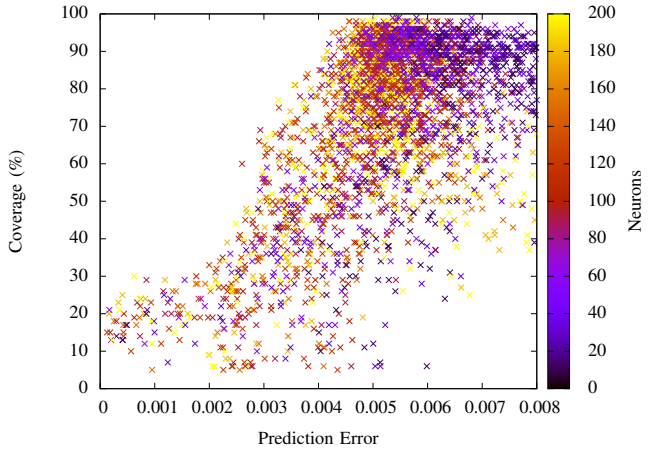


Figure 9: Scatterplot of coverage and prediction error in dependence of the number of reservoir neurons for the ESN model. While the prediction error is higher for smaller reservoirs, the coverage is less affected by the neuron number. All trials were conducted at the same terrain difficulty.

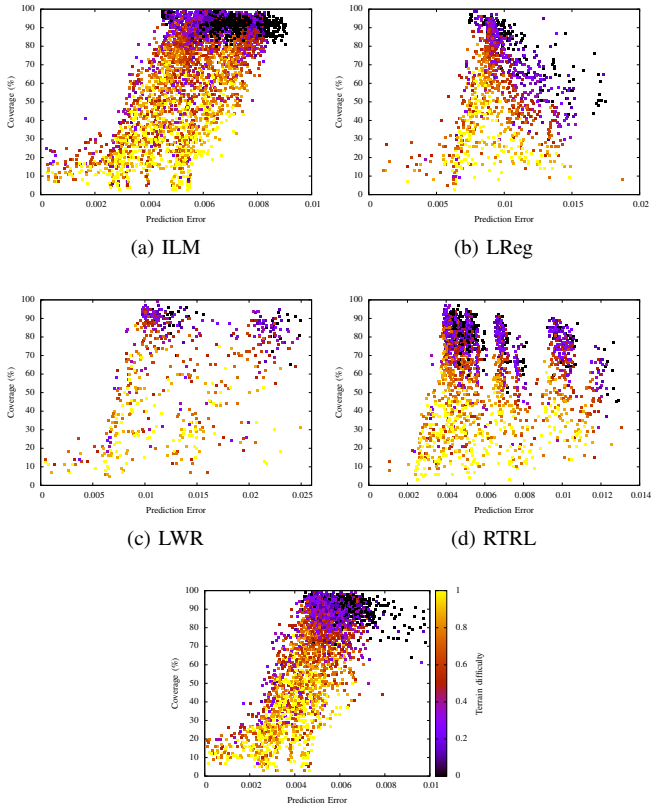
C. Model learning vs. behaviour learning

So far only the learning of the model ε_A (16, 17) was varied, while the learning rate of the controller ε_C (9) remained fixed. We now consider the ratio between the two learning rates $r = \frac{\varepsilon_C}{\varepsilon_A}$. Figure 8 shows coverage and prediction error for the ILM in a low difficulty environment as a function of the ratio between learning rates. Note that the optimal values between exploitation and exploration are found by test of different values from the parameter space. There is no on-line adaptation of learning rates.

Larger ratio implies a faster adaptation of the controller compared to the adaptation of the forward model, and vice versa. In Figure 8a, no explicit relation can be found about coverage. However, a faster adaptation of the model (smaller ratio) results in smaller prediction errors, see Figure 8b. The lowest values for the prediction error are achieved for the smaller ratio, increasing logarithmically with a linear ratio increase. With a small ratio, the model is able to adapt to the terrain fast enough to decrease the prediction error. From the point of view of the controller, it is not able to produce significantly new behaviour that would increase the prediction error. In a more difficult terrain, these tendencies remain but are less noticeable.

Different noise correlation (23) have been tested in order to compare the homeokinetic approach to a noisy signal. The best coverage performance is achieved at highly correlated noise (τ), and with smaller τ values almost all exploration is lost. When the noise is nearly independent the robot just *shake* in the position without producing any sensitive movement. Figures 4 and 5 show that the exploration achieved by the colour noise signal is worst when compared to any other model. These results present that a predictive and sensitive controller, e.g. homeokinesis, is able to induce a better exploration coverage and path length than a noisy signal.

The performance of the model-controller system depends



(a) ILM (b) LReg

(c) LWR (d) RTRL

(e) ESN

Figure 10: Prediction-coverage trade-off depends on the terrain difficulty. High coverage is usually achieved in simple environments. At higher difficulty, more structure is available and the robot has to change strategy often. The reduced space exploration leads to a reduction of the average error.

crucially on the complexity of the model. We have studied this relationship for the ESN which has turned out to perform generally very well in the tasks above. Different size in the reservoir, i.e. the number of neurons, affect the coverage and the prediction error. Trials have been taken with reservoir size ranging from 10 to 200 neurons. The results show that prediction error is higher for smaller reservoirs (Figure 9), and that coverage is less affected by the size of the reservoir.

IV. Discussion

A. Autonomous learning

Errors are the main source of information in most learning algorithms. While in machine learning, error-based information is typically used to calculate gradients with respect to a fixed data distribution, learning in autonomous agents is based on the effects from self-generated changes of the interaction with the environment, so that a constant data distribution cannot be assumed. Error-related information is incorporated into the IM in order to generate a standard to which later information is compared. The critical point in autonomous learning is not to reduce the error rate by learning, but to maintain a continuous stream of information. Practically,

autonomous learning is a dynamic competition between exploration (of the behavioural space) and exploitation (reduction of prediction errors). The exploration-exploitation dilemma is not to be pre-decided in terms of a "correct" setting of the meta-parameters of the learning algorithm, but needs to be understood as a balance which makes meaningful learning possible. The internal model transfers behaviourally caused and perceptually available information into a basis for the control of behaviour. The resulting behaviour is characterised by a continuum of options ranging from a combination of small prediction errors and little exploration, to large errors and ample exploration. Therefore, the evaluation of optimality of the model-based controller requires a Pareto principle to allow for either of these options. This is particularly evident if the environment is heterogeneous, i.e. containing simple and difficult regions, a case that is subject of further study.

While the Pareto frontier can guide the choice for the internal model, the specification of a point along the Pareto frontier reflects the properties of the environment, the availability of information, the spatial distribution and local dependency of information in nearby places, and the compatibility of the exploration strategy with the environment. It manifests itself only during runtime and within the interaction of the agent with the environment.

Information-theoretic approaches [12], [14] provide a theoretical approach to the characterisation of the Pareto-frontier by a single quantity, e.g. predictive information [31], [32], but require a probabilistic formulation of the problem that is problematic in the case of a single robot. In practical situations, it will be difficult to decide whether any detected information-theoretic suboptimality is due to an inefficiency of the agent or due to the poor quality of the estimation or approximation of the relevant entropies based on limited, non-stationary and statistically dependent data. The cost of obtaining information from either increased exploration or more precise local learning must also be taken into account. Such cost may not be comparable in practical problems as different time-scales, energy expenditure and risks are implied in either case. Nevertheless, in future work, it would be interesting to consider the informational properties of the internal models within the context of an information-theoretic description of the behaviour of the agent.

B. Implications of the experiments

We have chosen a simple robot and measured the degree of coverage of the planar layout of the environment in order to compare the robot's capabilities of exploration. In a more complex robot, various conditions need to be obeyed, such as remaining upright or avoiding self-blocking configurations which, for an exploratory problem, may lead to many different solutions[3]. Planar coverage provides us with a summary measure of the robot's ability to deal with and to switch between various slopes and indicates thus a degree of flexibility.

Given that learning takes time and requires a repeated encounter of the same situations, the reduction of errors tends to reduce the size of the explored region. On the other hand, large errors tend to have large effects on the controller such that less persistent behaviour can be expected. Although prediction error and coverage rate appear to be opposed criteria, it will turn out to depend on the model whether a good compromise can be found.

We have not aimed at a representation of the full layout of the environment or considered how the locally predictable situations that were identified by our set of models fit together. Such large-scale representations can be obtained by available simultaneous localisation and mapping (SLAM) algorithms based on exploratory behaviour of the robot. Our approach would provide a set of behaviour not assumed to be know to the robot. Thus, allowing accessibility to otherwise not reachable regions of the environment.

C. Effect of the type of model learning

Finally, we want to address the question which model performs best or, rather, what properties a model should have in autonomous learning. Obviously, the answer depends on constraints that are implied by the robotic hardware, by its computational resources or by the purpose of the robot. Also the software architecture is critical. If large-scale information is taken care of by high-level algorithms, a simple model will be sufficient if it represents information quickly and without biases towards outdated information. A more complex model, e.g. ESN (or it successor, the conceptor network [33], which can directly interface with symbolic algorithms), has potentially the representational power to cover a large part of the environment, although in this case learning times would be required that are longer than considered here. The internal models used in our experiments are able to learn only local representations of the environment. Due to this locality, optimum results are invariant to exploration time. Only ESN, that present an improved memory capacity compared to the other models, can benefit from longer explorations. Thus, a reduced prediction error can be achieved as the repeated state space has been already learned by the model.

V. Conclusion

Existing animal species have survived relying on brains with a wide range of sizes. Success in evolution appears to be independent of the absolute brain size as long as the complexity of the internal representations matches that of the survival-related aspects of the environment. This suggest that autonomous robots can be successful at various levels of computational power.

In an open environment, an agent shows a trade-off between accuracy of the representation and amount of represented information. In a finite memory capacity agent, if the information is homogeneously distributed in the environment, we expect a linear relationship which can be described as a Pareto frontier over models. Although a meta-criterion can select preferred points on the frontier, it may be useful to disambiguate for a specific environment rather than in advance in order to generate an optimal information flow from the environment to the agent's internal representation.

[3]The behavioural space of complex robots can be successfully explored in a similar way, as made evident by the examples at [30].

ACKNOWLEDGEMENT

We are grateful to Athanasios Polydoros for helpful comments and for the source code for some of the models. S.C.S. was funded by *The Advanced Human Capital Program of the National Commission for Scientific and Technological Research (CONICYT) of the Republic of Chile*.

REFERENCES

[1] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.

[2] M. Desmurget, C. M. Epstein, R. S. Turner, C. Prablanc, G. E. Alexander, and S. T. Grafton, "Role of the posterior parietal cortex in updating reaching movements to a visual target." *Nature Neuroscience*, vol. 2, no. 6, pp. 563–567, Jun. 1999.

[3] P. L. Gribble and D. J. Ostry, "Compensation for interaction torques during single- and multijoint limb movement," *Journal of Neurophysiology*, vol. 82, pp. 2310–2326, 1999.

[4] R. S. Johansson and G. Westling, "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects," *Experimental Brain Research*, vol. 56, pp. 550–564, 1984.

[5] D. Volkinshtein and R. Meir, "Delayed feedback control requires an internal forward model," *Biological Cybernetics*, vol. 105, no. 1, pp. 41–53, 2011.

[6] D. M. Wolpert and Z. Ghahrami, "Computational principles of movement neuroscience," *Nature Neuroscience*, vol. 3, 2000.

[7] D. E. Berlyne, "Curiosity and exploration," *Science*, vol. 153, no. 3731, pp. 25–33, 1966.

[8] J. Schmidhuber, "Curious model-building control systems," in *Neural Networks, 1991. 1991 IEEE International Joint Conference on*. IEEE, 1991, pp. 1458–1463.

[9] J. M. Herrmann, "Dynamical systems for predictive control of autonomous robots," *Theory in Biosciences*, vol. 120, no. 3-4, pp. 241–252, 2001.

[10] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE transactions on evolutionary computation*, vol. 11, no. 2, pp. 265–286, 2007.

[11] A. A. Feldbaum, "Dual control theory, Part I & II," *Automation and Remote Control*, vol. 21, no. 9 & 11, pp. 874–880 & 1453–1464, 1961.

[12] N. Ay, N. Bertschinger, R. Der, F. Güttler, and E. Olbrich, "Predictive information and explorative behavior of autonomous robots," *The European Physical Journal B*, vol. 63, no. 3, pp. 329–339, 2008.

[13] K. Zahedi, N. Ay, and R. Der, "Higher coordination with less controla result of information maximization in the sensorimotor loop," *Adaptive Behavior*, vol. 18, no. 3-4, pp. 338–355, 2010.

[14] D. Y.-J. Little and F. T. Sommer, "Learning and exploration in action-perception loops," *Frontiers in neural circuits*, vol. 7, p. 37, 2013.

[15] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "Vime: Variational information maximizing exploration," in *Advances in Neural Information Processing Systems*, 2016, pp. 1109–1117.

[16] R. Der and G. Martius, *The playful machine*. Springer, 2012.

[17] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.

[18] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International Conference on Machine Learning (ICML)*, vol. 2017, 2017.

[19] M. Ehrgott, *Multicriteria optimization*. Springer Science & Business Media, 2013, vol. 491.

[20] R. Der, "Self-organized acquisition of situated behaviors," *Theory in Biosciences*, vol. 120, pp. 179–187, 2001.

[21] J. Tani, R. Nishimoto, and R. W. Paine, "Achieving 'organic compositionality' through self-organization: Reviews on brain-inspired robotics experiments," *Neural Networks*, vol. 21, no. 4, pp. 584–603, 2008.

[22] O. Gigliotta, G. Pezzulo, and S. Nolfi, "Emergence of an internal model in evolving robots subjected to sensory deprivation," in *Proceedings of the 11th International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, ser. SAB'10. Berlin, Heidelberg: Springer, 2010, pp. 575–586.

[23] J. Schmidhuber, "Formal theory of creativity, fun, and instrinsic motivation," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 230–247, 2010.

[24] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314, no. 5802, pp. 1118–1121, 2006.

[25] W. W. Hager, "Updating the inverse of a matrix," *SIAM Review*, vol. 31, no. 2, pp. 221–239, 1989.

[26] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[27] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[28] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[29] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note'," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, 2001.

[30] G. Martius, "Lpzrobots simulator," http://robot.informatik.uni-leipzig.de/software, 2012.

[31] W. Bialek and N. Tishby, "Predictive information," *arXiv preprint cond-mat/9902341*, 1999.

[32] S. Still, "Information-theoretic approach to interactive learning," *EPL (Europhysics Letters)*, vol. 85, no. 2, p. 28005, 2009.

[33] H. Jaeger, "Controlling recurrent neural networks by conceptors," *CoRR*, vol. abs/1403.3369, 2014. [Online]. Available: http://arxiv.org/abs/1403.3369

APPENDIX

A. Controller update for a neural network predictor

Equation 9 can be written as

$$\Delta c = \varepsilon_C \boldsymbol{\chi}^\top \frac{\partial L}{\partial c} \boldsymbol{\eta}, \tag{24}$$

Using the auxiliary vector $\boldsymbol{\chi} = L^\top \boldsymbol{\eta}$, with $\boldsymbol{\chi} = (L^\top)^{-1}\boldsymbol{\eta}$ and c as a parameter of the controller [16]. To find the controller learning rules in dependence with each model first $L = \frac{\partial M}{\partial \mathbf{x}}$ and then $\frac{\partial L}{\partial c}$ have to be derived. An RNN as an extended model (dependence on the sensory input as well as in the motor commands) with no feedback and identity output function is defined as:

$$M(\mathbf{x}_t, \mathbf{y}_t) = \hat{\mathbf{x}}_{t+1} = W^{\mathrm{xo}}\mathbf{x}_t + W^{\mathrm{yo}}\mathbf{y}_t + W^{\mathrm{so}}\mathbf{s}_t.$$

In order to find

$$L = \frac{\partial M(\mathbf{x}_t, \mathbf{y}_t)}{\partial \mathbf{x}} = W^{\mathrm{xo}} + W^{\mathrm{yo}} G' C + W^{\mathrm{so}} \frac{\partial \mathbf{s}_t}{\partial \mathbf{x}}, \tag{25}$$

where G' is the same as in (12). The derivative depends on the network reservoir

$$\mathbf{s}_{t+1} = d(W^{\mathrm{xs}}\mathbf{x}_{t+1} + W^{\mathrm{ys}}\mathbf{y}_{t+1} + W^{\mathrm{ss}}\mathbf{s}_t)$$
$$\frac{\partial \mathbf{s}_t}{\partial \mathbf{x}} = D'(W^{\mathrm{xs}} + W^{\mathrm{ys}} G' C + W^{\mathrm{ss}} \frac{\partial \mathbf{s}_{t-1}}{\partial \mathbf{x}})$$

where, analogously to G', D' is the diagonal matrix $D'_{ij} = \delta_{ij} d'_i(\mathbf{w})$ with $\mathbf{w} = W^{\mathrm{xs}}\mathbf{x}_t + W^{\mathrm{ys}}\mathbf{y}_t + W^{\mathrm{ss}}\mathbf{s}_{t-1}$ and assuming $\frac{\partial \mathbf{s}_{t-1}}{\partial x} = 0$, the Jacobian matrix for the RNN is

$$L = W^{\mathrm{xo}} + W^{\mathrm{yo}} G' C + W^{\mathrm{so}} D' (W^{\mathrm{xs}} + W^{\mathrm{ys}} G' C). \tag{26}$$

Now the derivative of the Jacobian with respect to the matrix parameter C

$$\begin{aligned} \frac{\partial L}{\partial C} \quad = \quad & W^{\mathrm{yo}} \frac{\partial G'}{\partial C} C + W^{\mathrm{yo}} G' \\ & + W^{\mathrm{so}} D' W^{\mathrm{ys}} \frac{\partial G'}{\partial C} C + W^{\mathrm{so}} D' W^{\mathrm{ys}} G'. \end{aligned} \tag{27}$$

The activation function defined as $g(\cdot) = \tanh(\cdot)$ holds that $g'' = -2gg' = -2\mathbf{y}g'$ then we can derive

$$\frac{\partial G'}{\partial C} = -2\mathbf{y}G'\mathbf{x}.$$

Now defining and redefining the auxiliary vectors

$$\boldsymbol{\mu} = \boldsymbol{\chi}^\top W^{\mathrm{yo}} G', \;\; \boldsymbol{\upsilon} = \boldsymbol{\chi}^\top W^{\mathrm{so}} D' W^{\mathrm{ys}} G', \;\; \boldsymbol{\zeta} = C\boldsymbol{\eta}, \quad (28)$$

replacing term in (24) using (27) and (28), the learning rules for the parameters of the controllers are

$$\frac{1}{\varepsilon_C}\Delta C_{ij} = \mu_i \left(\eta_j - 2\zeta_i y_i x_j\right) + \upsilon_i \left(\eta_j - 2\zeta_i y_i x_j\right), \quad (29)$$

$$\frac{1}{\varepsilon_C}\Delta h_i = -2\mu_i \zeta_i y_i + 2\upsilon_i \zeta_i y_i. \quad (30)$$

The first term on the RHS of (29) and (30) represents the influence of the motor commands to the output, similar to the linear model (14, 15). The second term adds the influence of the reservoir with respect to the motor command to the output.

B. Response Model for RTRL

For a RTRL network, used as forward model for a homoekinetic controller, the response L has to be derived. The response is defined as:

$$L = \frac{\partial M\left(\mathbf{x}_t, K\left(x_t\right)\right)}{\partial x}. \quad (31)$$

For a RTRL network with activation function f and feedback weights W^{os}, the output is calculated as:

$$\hat{\mathbf{x}}_{t+1} = f\left(W^{yo}\mathbf{y}_t + W^{so}\mathbf{s}_t + W^{oo}\hat{\mathbf{x}}_t\right), \quad (32)$$

and the internal state update is:

$$\mathbf{s}_{t+1} = d\left(W^{ys}\mathbf{y}_{t+1} + W^{ss}\mathbf{s}_t + W^{os}\hat{\mathbf{x}}_t\right). \quad (33)$$

The derivation of L is as follows:

$$\frac{\partial M}{\partial \mathbf{x}} = F'\left(W^{yo}G'C + W^{so}\frac{\partial \mathbf{s}_t}{\partial \mathbf{x}}\right), \quad (34)$$

where G' as Equation 12 and F' the diagonal matrix:

$$F'_{ij} = \delta_{ij} f'\left(q_i\right) = \delta_{ij} f'_i\left(\mathbf{q}\right), \quad (35)$$

$$\mathbf{q} = W^{yo}\mathbf{y}_t + W^{so}\mathbf{s}_t + W^{oo}\hat{\mathbf{x}}_t. \quad (36)$$

The derivative of the internal state $\mathbf{s}$ with respect to the input is:

$$\frac{\partial \mathbf{s}_t}{\partial \mathbf{x}} = D'\left(W^{ys}G'C\right), \quad (37)$$

where D', analogously to G' and F' is the diagonal matrix defined as:

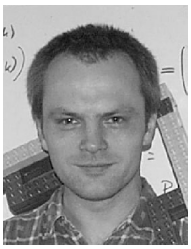
$$D' = \delta_{ij} d'(k_i) = \delta_{ij} d'(\mathbf{k}), \quad (38)$$

$$\mathbf{k} = W^{ys}\mathbf{y}_{t+1} + W^{ss}\mathbf{s}_t + W^{os}\hat{\mathbf{x}}_t. \quad (39)$$

Assuming $\frac{\partial \hat{\mathbf{x}}_{t-1}}{\partial \mathbf{x}_t} = 0$ and $\frac{\partial \mathbf{s}_{t-1}}{\partial \mathbf{x}_t} = 0$, the model response finally is:

$$L = F'\left(W^{yo}G'C + W^{so}\left(D'\left(W^{ys}G'C\right)\right)\right). \quad (40)$$

Simón C. Smith obtained his doctorate degree from the School of Informatics of the University of Edinburgh, UK. He obtained his bachelor and MSc from the University of Concepcion, Chile. Presently, he works as a research assistant at the Institute of Perception, Action and Behaviour at the University of Edinburgh. His main focus of research includes explainable AI, reinforcement learning, intrinsic motivations for autonomous robots, and information theory.



J. Michael Herrmann received his PhD in Computer Science from Leipzig University, Germany, in 1993. He has been a Postdoctoral Researcher at Leipzig University, at NORDITA, at RIKEN's Information Representation Lab, and at the MPI for Dynamics and Self-Organisation at Göttingen. Presently, he is a lecturer at the School of Informatics, University of Edinburgh. His research interest include neural networks, the theory of dynamical systems, optimisation algorithms and learning in autonomous robots.